An Approach for Detecting Image Spam in OSNs

Niddal H. Imam
Department of Computer Science
University of York
York, UK
ni571@york.ac.uk

Vassilios G. Vassilakis
Department of Computer Science
University of York
York, UK
vasileios.vasilakis@york.ac.uk

Abstract

In recent years, the number of images uploaded into Online Social Networks (OSNs), such as Facebook and Twitter has been growing, which presents challenges to Machine Learning-based spam detector. Most current detection models use text-based, statistic info-based and graph-based features can easily be fooled by image-based spam. These approaches do not have the ability to recognize text embedded in images. Adversaries take advantage of this issue to launch more sophisticated attacks, such as evasion attacks. Thus, this paper proposes an adversary-aware model for detecting spam images in OSNs. The proposed model adopted EAST (an Efficient and Accurate Scene Text Detector) and CRNN (Convolutional Recurrent Neural Network) models for text detection/ recognition tasks. After the text extraction step, a black-list and white-list with Human-in-the-loop approach is applied for text classification task. Although the technique used is simple, it is adaptable and robust against adversarial text attacks.

1 Introduction

The amount of information shared in OSNs has continued to increase in recent years. One study shows that the number of profiles on Facebook, Twitter, and LinkedIn reached more than two billion in 2016 (Ala'M et al., 2017). Facebook, the most popular OSN in the world, had 1.87 billion monthly active users and 1.15 billion daily active users during the period of January to February 2017 (Watcharenwong and Saikaew, 2017). A study reported that over the course of one month, Twitter has two million users sharing 8.3 million tweets per hour (Mateen et al., 2017).

Unfortunately, the high popularity of these OSNs has made them very attractive to malicious users, or spammers. Spam is referred to as an unsolicited message that is received from a random sender with no relationship to the receiver. These messages may contain malware, advertisements, or URLs directing the recipients to malicious websites (Barushka and Hajek, 2018). Although spam is prevalent in all forms of online communication (such as email and the web), researchers and practitioners attention has increasingly shifted to spam in OSNs, due to the growing number of spammers and the possible negative effects on users (Barushka and Hajek, 2018) (Sedhai and Sun, 2015).

Moreover, current spam detectors that use text-based, statistic info-based and even graph-based features can easily be fooled by image-based spam, where an adversary inserts text inside an image. A recent survey conducted by (Wu et al., 2018) presented the pros and cons of these detectors and suggested of developing a comprehensive model to improve detection performance. (Biggio et al., 2011) reported that although spam-based images have been widely used in email since 2006, no proper defence strategy has yet been developed. Spam images have been replaced by URL-based spam, as the latter requires a lower email size and is able to send more messages. However, the volume of images being shared in OSNs has been growing, partly due to the increase in network bandwidth. Processing large numbers of images and the retrieval of textual content from images are challenges in OSNs. Although some of the OSNs' platforms, such as Twitter has provided Muting options to enable users to block Tweets contain particular words or hashtags, these Muting options can not block image-based spam. Figure 1 shows some examples of spam images found in Twitter. Setting your Twitter account to mute words, such as 18+, or MULTI MACA, will not block the examples in Figure 1. Thus, a solution that considers extracting text from images is therefore needed.

Recent work has developed an ML model that uses Optical Character Recognition (OCR) to de-

Figure 1: Examples of image-based spam.

tect spam-based images on Facebook (Borisyuk et al., 2018). According to (Smith, 2013) the number of images uploaded to Facebook is now in the hundreds of millions. Spammers take advantage of images on OSNs by inserting malicious content into images to evade detection. One technology that enables both handwritten and printed text to be extracted from images is OCR. Although OCR has shown some weakness in the past, seminal work by (LeCun et al., 1995) and other advances in deep learning for object recognition tasks (e.g., CTC, and ConvNet) has helped to overcome some of these drawbacks (Graves et al., 2006) (Delakis and Garcia, 2008). These works use DNNs as a feature extractor, which allows use of variable-size image inputs (Song and Shmatikov, 2018).

Common methodologies used in the literature for classifying text are based on Natural Language Processing (NLP) techniques, such as Word2vector and Bag-of-Word (BoW), or Multilayer perceptron (MLP). However, recent studies have shown that these techniques are vulnerable to malicious activities, enabling an adversary to mislead deployed models. Adversaries can easily fool NLP models by adding, removing or replacing words (Samanta and Mehta, 2017). Also, as cited in (Eger et al., 2019), the lack of robustness to morphological variation or spelling mistakes can be exploited as a blind spot of NLP techniques. These examples show the distinction between human and machine learning models. Human intervention is important to defend against adversarial attacks.

Thus, this paper proposes a pipeline that uses a deep learning approach for extracting text from spam images. Then, a blacklist/ white-list approach with human assistance is applied for classifying extracted words. Also, the proposed pipeline is designed to detect new or modified words (adversarial examples).

The remainder of this paper is structured as follows: Section 2 provides an overview of previous research on image spam detection. Section 3 describes the methodology used for detecting, recognizing and classifying image spam in OSNs. Experimentation and evaluation of the proposed model is discussed in Section 4. The conclusion and future works are presented in Section 5.

2 Related Works

There are two types of image spam detection: an image content-based and characteristics-based. The first type attempts to extract the text in a spam image and then make detection decisions. Similarly, it would follow the same process with non-image spam. On the other hand, the second type attempts to detect image spam based on the characteristics of image files. This paper pursues the first approach, but first, a revision of some relevant related work for characteristics-based approaches is going to be discussed. In (He et al., 2016) the authors develop a comprehensive model for detecting different types of spam in OSNs, including spam image. The developed model can detect images based on the following features: Colour and Edge Directivity Descriptor (CEDD), Gabor features, edge histograms and the Scale Invariant Feature Transform (SIFT). Also, (Annadatha and Stamp, 2018) proposed an approach for detecting spam images based on a set of images characteristics. The proposed approach extracts 21 features, such as colour, edges, comp, noise, etc., for training a linear Supervised Victor Machines (SVM) classifier. Two different datasets were used: a standard dataset and an improved dataset (more challenging). Although the prediction accuracy of the standard dataset reaches 95, the model was not capable of distinguishing between spam and non-spam images accurately when applied to improved dataset.

A recent study by (Borisyuk et al., 2018) developed an Optical Character Recognition (OCR) system that detects and recognizes text in images uploaded to Facebook. The system, called Rosetta, consists of two models: text detection and text recognition. The text detection model uses Fast-RCNN to perform word detection. It detects the locations of words in an image and produces words surrounded by bounding boxes. Then, for each detected box, a fully-convolutional model, referred to as CTC, is used to recognize text. The recognition model predicts the most likely character at each detected box in the image. For experiments, different datasets were used; a synthetic dataset was used for pre-training and COCO-text, and human rated datasets were then used for fine-tuning the models. The developed models were implemented in Detectron, an open-source software used for object detection research. Also, (Yuan et al., 2019) developed a model called Malena, which can detect different types of spam including images carrying text, number, or QR code in Chinese social networks (Baidu Tieba and Sina Weibo). The authors use an off-the-shelf tool, PixelLink (Deng et al., 2018) for detecting text in images.

Attacks against text classification models have been widely studied in the literatures. In (Li et al., 2018) spaces between words are replaced with special characters, such as hyphens or asterisks to fool word2vector models. Also, (Eger et al., 2019) investigated the robustness of state-of-the-art Deep Learning models against visual attacks, in which an adversary exchanges some characters with alternative visually similar ones (i.e. V1agra). These attacks are common in OSNs as they do not require any knowledge about the deployed model, nor any linguistic knowledge or and human understanding.

3 Methodology

The methodology used for building image spam detector involves three independent steps: text detection, recognition, and classification. In the first step, text regions in the spam image are detected. Second, the detected text regions are recognized and saved as text file. In the last step, the recognized words are classified as either spam or non-spam. Figure 2 illustrates the architecture of the overall model.

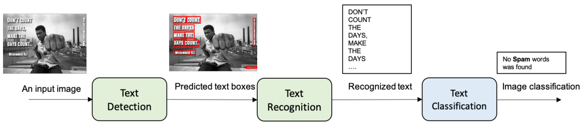

Figure 2: Overall model architecture. The model consists of three steps: text detection based on EAST, word recognition using CRNN, and a proposed text classification step.

3.1 Text Detection Model

The model used for text localization is called EAST (Efficient and Accurate Scene Text), as proposed in (Zhou et al., 2017). The model has been widely used in the literature (Long et al., 2018) (Liao et al., 2018). The model consists of a few stages, including a fully convolutional network (FCN) and Non-maximum suppression (NMS), unlike some of the existence models. The FCN takes an image and produces multiple channels of text score map and geometry. One of the predicted channels is a score map, and its pixel values are [0,1]. The remaining channels belong to geometries, which crop the word from the view of each pixel. Two geometry shapes are used for text regions: rotated box (RBOX) and quadrangle (QUAD). After calculating the loss functions for each, thresholding is applied to find the geometries that have scored over the predefined threshold. Those geometries are considered valid.

3.2 Text Recognition Model

An approach based on convolutional recurrent neural networks (CRNN) proposed in (Shi et al., 2017) was adopted. The CRNN model is a combination of Deep Convolutional Neural Network (DCNN) and Recurrent Neural Network (RNN). The architecture of CRNN consists of three components: Convolutional Layers, Recurrent Layers, and Transcription Layer.

Convolutional layers are constructed to extract sequential features from an input image. The input image is divided into columns from left to right. Each feature map column corresponds to a rectangular region of input image. Thus, each vector feature of a feature sequence is considered as a descriptor of that rectangular region. Then, the recurrent network is used to make predictions for the output of the convolutional layers, which is a set of feature sequence frames. A label distribution for each frame in the feature sequence x is predicted by the recurrent layers. One of the advantages of recurrent layers is that RNN can capture contextual information within a feature sequence. As the traditional RNN suffers from a vanishing gradient problem, Long-Short Term Memory (LSTM) was used. LSTM consists of memory cells to store the past context, and input/output gates to store context for a long period of time. The third component of LSTM is forget gates, which are used to clear the memory cell. A deep bidirectional LSTM, which consists of forward and backward LSTMs, was used. The transcript layer translates the per-frame predictions of the recurrent layers into a label sequence. It finds the label sequence that has the highest probability conditioned among the pre-frame predictions. This layer uses Connectionist Temporal Classification (CTC) for the conditional probability task. The conditional probability of all sequences is defined using the following equation Eq. 1:

$$p(l|y) = \sum\nolimits_{\pi:\mathcal{B}(\pi)=1}^{n} p(\pi|y)$$

There are two transcription modes that can be used: lexicon-free and lexicon-based transcripts. In lexicon-free mode, the probability of sequences that are calculated by using Eq. 1 are taken as the predictions. However, in lexicon-based mode, the results of Eq. 1 are associated with lexicon, which is a spell-checking dictionary.

3.3 Text Classification Model

The text classification task consists of two steps: text pre-processing and classification result. A Natural Language Toolkit (NLTK) package was used for text pre-processing. First, extracted text from an image is steamed to convert words to their base forms. Then, a simple approach is used to classify the pre-processed text. As the output of the text recognition is a set of words, a blacklist of spam words is used. The blacklist used in this paper was created in 2011 and is updated regularly, it collected from Wordpress comments. Additionally, the NLTK package was used to build a whitelist on non-spam words, which helps to detect a new word that might be used by an adversary. Consequentially, if a spam word is detected, the model will notify a user that this image is a spam image, or if a new word that cannot be found in neither the blacklist nor the whitelist is detected, the new word will be sent to an admin for conformation. The admin tasks are classifying new detected words and updating the lists. A flowchart for text classification process is illustrated in Figure 3.

4 Experiments

4.1 Models

A dataset of images with embedded text were collected from Twitter trending hashtags and topics. The collected images contain text with different fonts, sizes and language. The collected dataset was used to choose the best text detection model. Three text detection models, which are publicly made available, were evaluated: EAST (Zhou et al., 2017) Connectionist Text Proposal Network (CTPN) (Tian et al., 2016) and You only look once (YOLOv3) (Redmon and Farhadi, 2018). Figure 4 shows the output of the three models when using an OSNs image.

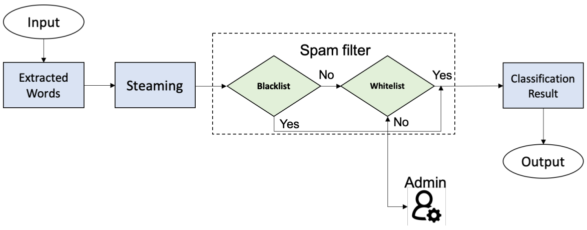

Figure 3: Text Classification Process.

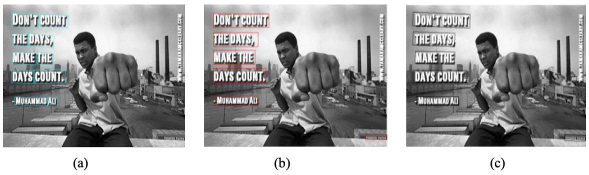

Figure 4: This figure shows the output of three text localization models: (a) EAST; (b) CTPN, (c) YOLOv3.

EAST was chosen for several reasons: it was written in Keras, which is easy to read and run. It can detect a single word unlike CTPN and Yolov3 models, which detects several words (see Figure 4). Detecting a single word is important as CTC separates characters by blanks, so it cannot separate words in a sentence. Also, detection time per image was compared and EAST was found to be the fastest among the three models (see Table 1). Also, it has been reported in (Deng et al., 2018) that EAST runs fast.

Images	EAST	CTPN	YOLOv3
1	0.66 s	2.95 s	1.84 s
2	0.63 s	2.94 s	1.92 s
3	0.64 s	3.10 s	1.90 s
4	0.63 s	3.18 s	1.88 s
5	1.43 s	3.22 s	1.80 s

Table 1: Cost Time a model take per second.

4.2 Datasets

Different datasets were used for training and evaluating the proposed pipeline. A public dataset built for ICDAR2017 Competition on Multilingual scene text detection and script identification (Nayef et al., 2017) was used for training the text detection model. The dataset is a collection of natural scene images with embedded text. It consists of 18,000 images containing text, such as street signs, advertisement boards and shops names, for 9 languages and symbols. This enables it be used as a benchmark for testing algorithms ability to distinguish different scripts in images. Although it does not match images found in OSNs exactly, it resembles some of OSNs images characteristics, such as languages, fonts, natural scene backgrounds, and text locations and directions. OSNs images have a combination of the following characteristics: variation of aspect ratio, multi-oriented, curved text, variation of fonts, and multilingual text. In addition, a dataset of images with embedded text was manually collected and annotated from Twitter to fine-tune the model.

Several datasets were used for building the text recognition model. The synthetic word dataset was used for training the CRNN model (Jaderberg et al., 2014). Also, a synthetic Arabic dataset was created using a framework for generating synthetic datasets that do not require human-labelling. Cropped images with embedded Arabic words were generated by using a list of 15 thousand Arabic words.Moreover, A dataset created for ICDAR2017 competition on cropped word recognition was used. The dataset consists of 2000 samples with embedded English and Arabic text. Additionally 1000 samples were collected from Twitter.

For the text classification task, the Wordprees comments dataset and SMS spam dataset built by (Almeida et al., 2011) were used to create both the blacklist and white-list. The Wordpress dataset contains 36,000 phrases, patterns, and keywords. The SMS spam dataset, which is commonly used in literature for building spam detection models, contains a set of 5,574 SMS massages classified as spam or ham. The list of spam words was used as a blacklist, while Ham messages corpus was extracted to build the white-list. Both lists will be updated regularly as spam words used in OSNs might be different.

4.3 Training

Some recent studies proposed End-to-End OCR models that use a single dataest for training both text localization and recognition. However, in this paperer, the pipeline is trained in a two-step fashion, where each part of the pipeline is trained separately. A two-step process model has some advantages, including the ability to update a single part of the model when it is needed. Most importantly, this process ensures that if part of the model is been compromised, other pats of the model would not be affected.

Text localization model. The adopted EAST model uses AdamW (Loshchilov and Hutter, 2019) optimizer, to speed up learning process. Also, the adopted EAST model uses ResNet-50 (He et al., 2016) as a backbone instead of PVANet as used in the original structure of the EAST model. A pre-trained model, trained on ICDAR 2013 and 2015 benchmarked datasets, was used. The pre-trained model achieved an 0.802 F-score on ICDAR 2015 test dataset. ICDAR 2017 and the developed Twitter datasets were used for fine-tuning the model. Detection accuracy results are discussed in the following sections.

Text recognition model. A combination of synthetic and cropped images with embedded English and Arabic text were used for training and evaluating the model. 7,493,000 training and 943,740 validation samples were used to build a new text recognition model. The model was trained for 25 epochs and it achieves 0.83 accuracy on the validation dataset.

4.4 Evaluations and Results

Results of the text localization and text recognition models were evaluated separately. Different metrics were used for evaluating the models. Metrics used for evaluation were adopted from (Karatzas et al., 2015).

Text localization model. First a dataset of images with embedded text was collected from Twitter to test the text localization model as there is not an OSNs benchmarked dataset. 300 images with embaded text were collected from Twitter, and the dataset were split into 200 training, 50 validation and 50 test images. ICDAR 2013 + 2015, ICDAR 2017, and Twitter datasets were used to train the text localization model. After that, the three models were tested by using Twitter test dataset. Table

2 shows the evaluation results of the three models that trained on different datasets. The model fine-tuned by 200 OSNs dataset shows an overall improvement in the performance of the detection better than the other two models.

Metrics used for evaluating the models are: Precision, Recall, and F-measure (F1 score), that is the harmonic mean of precision and recall. These metrics were defined in the ICDAR 2013-2015 challenge. After annotating the test dataset collected from Twitter, the ground truth were compared with the result of each model.

No.	Model	precision	recall	hmean
1	ICDAR 2013 + 2015	0.65	0.60	0.62
2	ICDAR 2017	0.78	0.65	0.71
3	ICDAR 2017 + Twitter	0.79	0.68	0.73

Table 2: Evaluation of the text localization model using models trained on three different datasets.

Text recognition model. The dataset collected form Twitter to test the text localization model was cropped to be used for evaluating the text recognition model. The total number of samples used to test the model is 120 images with embedded Arabic and English text. The performance of the model in recognizing Arabic and English text were compered. Table 3 presents the results of the text recognition model using Twitter test dataset.

The code adopted for evaluating the model were built by (Karatzas et al., 2015) in Incidental Scene Text 2015 competition. The metrics used for evaluation are: (CRW) Correctly Recognized Words, and (TED) Total Edit distance. Table 3 shows the CRW for English samples is higher than the Arabic one, which means that the model can recognize English text better than Arabic text. One of the reasons is that the number and quality of English language samples used for training the model is higher than the Arabic ones.

No.	Language	No. of Samples	TED	CRW
1	Arabic	102	166.0	0.460
2	English	49	51.0	0.571
3	Arabic + English	121	199.0	0.305

Table 3: Evaluation of the text recognition model using Twitter test dataset.

Text classification model. As this model is highly dependent on the output of the text localization/recognition models, no evaluation has been carried out for this model. The model uses black and White lists that updated whenever a new word is detected. Thus, the detection accuracy of the text localization/recognition is crucial.

Ethical issue. Images collected from Twitter were posted by users in trending hashtags. Users account from which these images were collected have not been analysis in this research. Thus, this paper is not Involving Human Subjects. Also, some examples for images collected from Twitter were presented in this paper to help the readers understanding the problem that this paper is trying to solve.

4.5 Discussion

The evaluations and results section shows that the overall performance of the pipeline need to be improved. Although the text localization part of the pipeline achieves 0.73 detection accuracy on Twitter dataset, the performance of text recognition part of the pipeline need to be improved. More Arabic samples need to be collected to improve the text recognition model. Also, one of the issues that has been found when building model for recognizing Arabic and English text is that the model mistakenly recognize some Arabic letters that have shapes close to some English letters' shape.

Recent studies have shown when deploying DL-based model for security application, they become vulnerable to different adversarial attacks. Consequently, two security countermeasures were taken into account when designing the proposed approach. First, the pipeline is trained in a two-step fashion,where each part of the model is trained separately. Unlike end-to-end models that use a single training dataset to train the model. This training process will ensure that if part of the model is attacked, the other parts would not be affected. Also, the model robustness against adversarial text attacks, where adversaries modify spam words to avoid detection was considered. A blacklist/ whitelist with Human-in-the-loop approach was proposed to detect modified or crafted words (adversarial examples).

Moreover, the proposed text classification model can be used for several purposes. For example, it can be used for training text recognition models. New or modified words can be used to update the CRNN and enable it to recognize new words. Also, it can be used as a defence method against one of the adversaries attack, in which an adversary contaminates training data to cause misclassification (causative attack). The proposed approach can be used to filter out contaminated samples that may be injected into the collected data for training ML models. Additionally, the proposed model can be used as either a Server-based or Client-based filter. Moreover, the proposed approach ensures that the deployed detection model is evolving and updated regularly, which is a very important defence strategy against adversaries.

5 Conclusion and Future work

An approach for detecting spam images in OSNs through localizing and recognizing text embedded in an image was presented. Text detection and recognition models that are commonly used in the literature were adopted, and a new method for classifying extracted text from images was proposed. The proposed semi-automated model was designed to be robust against adversarial text attacks.

In terms of limitations and future works, a couple of points need to be improved in the text detection model, such as detecting the maximum text size and detecting differently oriented text. Also, recognizing multiple language text is an area that needs to be improve in this model. As the text classification model proposed in this paper is based on blacklist, it may classify non-spam massages as spam (false positive) due to finding a spam word regardless of message context. A possible solution for this drawback could be examine another characteristic of a spam massage along with the appearance of spam words in the image. For example, if a single spam word is detected in an image, account features (e.g. number of friends, or account reputation) and the messages content features (e.g. number of hashtags, or number of words) need to be checked. Another possible solution is to notify users by hiding the spam massage and show the detected spam or modified word, so the user can make the judgment. However, future work will be focused on improving the text classification part of the model.

References

Al-Zoubi Ala'M, Hossam Faris, et al. 2017. Spam profile detection in social networks based on public features. In *2017 8th International Conference on information and Communication Systems (ICICS)*, pages 130–135. IEEE.

Tiago A Almeida, José María G Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 259–262. ACM.

Annapurna Annadatha and Mark Stamp. 2018. Image spam analysis and detection. *Journal of Computer Virology and Hacking Techniques*, 14(1):39–52.

Aliaksandr Barushka and Petr Hajek. 2018. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. volume 48, pages 3538–3556.

Battista Biggio, Giorgio Fumera, Ignazio Pillai, and Fabio Roli. 2011. A survey and experimental evaluation of image spam filtering techniques. volume 32, pages 1436–1446.

Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. 2018. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 71–79. ACM.

Manolis Delakis and Christophe Garcia. 2008. text detection with convolutional neural networks. In *VISAPP (2)*, pages 290–294.

Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. 2018. Pixellink: Detecting scene text via instance segmentation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding nlp systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 369–376, New York, NY, USA. ACM.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

M Jaderberg, K Simonyan, A Vedaldi, and A Zisserman. 2014. Synthetic data and artificial neural networks for natural scene text recognition. In *Neural Information Processing Systems*.

Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. 2015. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE.

Yann LeCun, Yoshua Bengio, and Others. 1995. Convolutional networks for images, speech, and time series. volume 3361, page 1995.

Yue Li, Pengjian Xu, and Minmin Pang. 2018. Adversarial attacks on word2vec and neural network. In *Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence*, page 50. ACM.

M Liao, B Shi, and X Bai. 2018. TextBoxes++: A Single-Shot oriented scene text detector. volume 27, pages 3676–3690.

Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. 2018. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *ECCV*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

M Mateen, M A Iqbal, M Aleem, and M A Islam. 2017. A hybrid approach for spam detection for twitter. In *2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 466–471.

N Nayef, F Yin, I Bizid, H Choi, Y Feng, D Karatzas, Z Luo, U Pal, C Rigaud, J Chazalon, W Khlif, M M Luqman, J Burie, C Liu, and J Ogier. 2017. ICDAR2017 robust reading challenge on Multi-Lingual scene text detection and script identification - RRC-MLT. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1454–1459.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. arXiv preprint arXiv:1707.02812.

Surendra Sedhai and Aixin Sun. 2015. HSpam14: A collection of 14 million tweets for Hashtag-Oriented spam research. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 223–232. ACM.

Baoguang Shi, Xiang Bai, and Cong Yao. 2017. An End-to-End trainable neural network for Image-Based sequence recognition and its application to scene text recognition. volume 39, pages 2298–2304.

Cooper Smith. 2013. Facebook users are uploading 350 million new photos each day. volume 18. Business insider.

Congzheng Song and Vitaly Shmatikov. 2018. Fooling ocr systems with adversarial text images. volume abs/1802.05385.

Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. 2016. Detecting text in natural image with connectionist text proposal network. volume 9912. Springer, Cham.

N Watcharenwong and K Saikaew. 2017. Spam detection for closed facebook groups. In *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6.

Tingmin Wu, Sheng Wen, Yang Xiang, and Wanlei Zhou. 2018. Twitter spam detection: Survey of new approaches and comparative study. volume 76, pages 265–284.

Kan Yuan, Di Tang, Xiaojing Liao, XiaoFeng Wang, Xuan Feng, Yi Chen, Menghan Sun, Haoran Lu, and Kehuan Zhang. 2019. Stealthy porn: Understanding real-world adversarial images for illicit online promotion. In *Stealthy Porn: Understanding Real-World Adversarial Images for Illicit Online Promotion*, page 0. IEEE.

Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. EAST: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560.